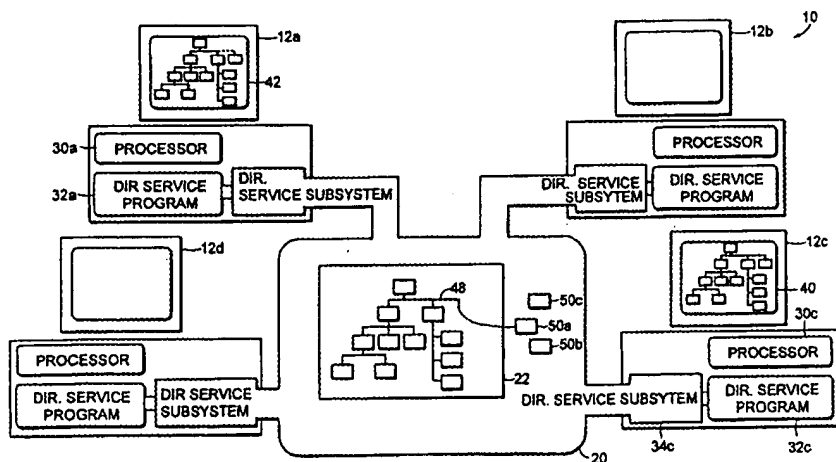




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> :  G06F 17/30		A1	(11) International Publication Number:  WO 98/22893
			(43) International Publication Date:  28 May 1998 (28.05.98)
(21) International Application Number:  PCT/US97/21733		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date:  21 November 1997 (21.11.97)			
(30) Priority Data: 08/754,481 22 November 1996 (22.11.96) US 08/827,534 28 March 1997 (28.03.97) US 08/850,137 2 May 1997 (02.05.97) US			
(71) Applicant: MANGOSOFT CORPORATION [US/US]; Suite 190, 1500 West Park Drive, Westborough, MA 01581 (US).		Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	
(72) Inventors: DIETTERICH, Daniel, J.; 4 Cedar Terrace, Acton, MA 01720 (US). PHILLIPS, Robert, S.; 5 Sherman Street, P.O. Box 601, Brookfield, MA 01506 (US). CARTER, John, B.; 414 South Douglas Street, Salt Lake City, UT 84102 (US). DAVIS, Scott, H.; 136 Riverbend Road, Groton, MA 01450 (US). FRANK, Steven, J.; 6 Tiffany Trail, Hopkinton, MA 01748 (US). ABRAHAM, William; 15 Galway Road, Windham, NH 03087 (US).			
(74) Agent: LANZA, John, D.; Testa, Hurwitz & Thibault, LLP, High Street Tower, 125 High Street, Boston, MA 02110 (US).			

(54) Title: DYNAMIC DIRECTORY SERVICE



**(57) Abstract**

In a network of computer nodes, a directory service provides both the physical location of directory information around the network and the directory information itself in a single data structure.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

## DYNAMIC DIRECTORY SERVICE

### Cross-Reference to Related Applications

This application is a continuation-in-part of co-pending U.S. patent applications serial number 08/754,481, filed November 22, 1996, and serial number 08/827,534, filed March 28, 1997 and bearing attorney docket number CLC-002. The entirety of both of these applications is  
5 incorporated herein by reference.

### Technical Field

The present invention relates in general to directory services and, more specifically, to a dynamic directory service that maintains a directory in which is stored both (1) directory information and (2) information identifying the physical layout or structure of the directory.

10

### Background Information

Computer-based structured storage systems, such as computer file systems and database systems, have been remarkably successful at providing users with quick and facile access to enormous amounts of data. Structured storage systems have allowed businesses to generate and maintain enormous stores of persistent data that the company can modify and update over the  
15 course of years. For many companies, this persistent data is a valuable capital asset that is employed each day to perform the company's core operations. The data can be, for example, computer files (e.g., source code, wordprocessing documents, etc.), database records and information (e.g., information on employees, customers, and/or products), and/or Web pages.

A typical computer based structured storage system includes a central server that  
20 provides centralized control over the structured store of data. The structured store of data is the information that is being maintained by the system, such as information in the files and directories of a file system or within the records of a database system. The central server provides system services to a plurality of interconnected network client nodes, and each of the client nodes employs the central server to access and manipulate the structured store of data.

25 It is common to use the central network server to provide a directory service, i.e., a specialized hierarchical database of network, user, and other computer system configuration information. This information typically includes operating system configuration information,

application program configuration information, network configuration information, network-accessible resources, exported devices and services, network printers, and user account records. Network user account records are what allow network-wide, unified "log on," and they allow sophisticated application programs to obtain broader, coherent name spaces (e.g., electronic mail system application programs). Novell's Directory Server (NDS), available from Novell, Inc. of Provo, Utah, is one example of software that implements a directory service and that runs on a central server to allow maintenance of and access to directory information.

Although central server directory services such as NDS generally work, problems arise from relying on centralized control. For example, the operation of the network is dependent upon the proper functioning of the central server. Any failure of the server to maintain proper operation, such as a power failure, hardware failure, or other such system failure, will disable the entire network and generally prevent users from obtaining access to the network and its resources. Additionally, a flood of client requests (e.g., access of user account records) can overload the central server and slow down or crash the network. Accordingly, reliance on a centralized, server-based directory service can result in slow operation or total network failure during periods of heavy use.

An additional problem with a client-server network system is that it provides a static operating environment that is set for optimal performance at a certain level of network activity. Consequently, the network fails to exploit available resources to improve system performance. In particular, as the system activity rises above or drops below the expected level of network activity, the static operating environment lacks any ability to reconfigure dynamically the allocation of network resources to one providing better performance for the present level of activity.

Technology has been developed to improve the reliability and operation of a centralized server directory service. This technology involves employing a plurality of central servers. Each of the servers provides a directory service. Whenever the directory information changes, all of the redundant servers are updated so that requests can be made to any of the centralized servers without impacting the correctness of the response.

While this statically replicated technology improves upon the single server arrangement, it generally performs poorly or fails as the size of the network increases. As client nodes are added

to the network and more directory servers are added to the network to handle the corresponding increased network load, the sustainable update rate of information managed by the servers decreases because the overhead and complexity of propagating updates (i.e., changes to the data stored in and replicated across the directory servers) increases with the increased network size.

- 5 Thus, known directory services are inherently read-mostly repositories, and that is the way they are designed and used. That is, known directory services are used only for infrequently-updated directory information.

- Examples of this centralized, statically replicated, hierarchical directory service technology include Microsoft's Active Directory (NT 5.0), NDS, Banyan's Streetalk, and X.500  
10 directory services. Lightweight Directory Access Protocol (LDAP) is a "common" protocol that can be used to access data from any compatible directory server such as NDS.

Summary of the Invention

It is an object of the invention to provide a dynamic directory service that is an improvement over centralized, statically replicated, hierarchical database directory services.

It is also an object of the invention to provide a dynamic directory service that maintains a  
5 directory in which is stored both (1) directory information and (2) the physical layout of the directory itself. That is, the invention involves the use of a directory having both information about where to find the directory information of interest as well as the actual directory information.

It is a further object of the invention to provide a dynamic directory service that is more  
10 reliable and provides improved fault tolerant operation over existing directory services, and that has the ability to replicate and move data dynamically in response to network activity levels and access patterns. This ability optimizes performance and minimizes the time required to provide directory information (e.g., network configuration information, organizational information, user configuration information, and network-accessible resources such as exported devices and  
15 services, network printers, and user account records) to requesting network nodes.

It is yet another object of the invention to provide a dynamic directory service that provides distributed control over a structured store of directory information and that allows that information to be changed and/or updated with higher frequency than generally possible with existing directory services without adversely effecting network performance or network node  
20 access times.

It is still another object of the invention to provide a dynamic directory service that maintains and allows access to frequently-changing, as well as infrequently-changed, directory information.

The directory service of the invention maintains both the physical location information  
25 and the directory information itself in the same data structure. That data structure is distributed around the network, and all of the directory information is homeless (except, usually, a root record). The distributed directory service of the invention allows network nodes to locate copies of records to which the nodes want access. A globally unique identifier (GUID) is associated with each record as a unique index key that can be used to identify the record on the network. It  
30 contains no location information, and a given record can be physically located anywhere on the

network. Each of the records also can migrate from physical location (node) to physical location (node) around the network. The invention provides automatic migration and/or replication of directory information among the network nodes without reconfiguring the network nodes and in a manner transparent to users at the nodes. An example of a mechanism that can be used to keep replicated records consistent is a single-writer, multiple-reader, write-invalidate protocol.

The directory service of the invention can, in some specific embodiments, employ a globally-addressable unstructured memory system to maintain simultaneously both the directory information and layout information in accordance with the invention. For example, the directory service can employ the distributed shared memory (DSM) system described in the above-identified, incorporated-by-reference patent applications, which DSM system distributes the storage of data across some or all of the storage devices connected to a network. Storage devices that may be connected to the network and accessible to the network nodes by address via the DSM system include, for example, RAM, hard disk drives, tape drives, floppy disk drives, and CD-ROM drives. In some embodiments, the dynamic directory service is a computer program that interfaces to a DSM system to operate the DSM system as a memory device that provides storage of and access to the directory information. The directory service program can direct the DSM system to map directory information into the shared memory space. The DSM system can include functionality to share, migrate, and replicate data coherently. In one embodiment, the DSM system provides memory device services to the directory service program. These services can include read, write, allocate, flush, or any other similar or additional service suitable for providing low level control of a storage device. The directory service program employs these DSM system services to allocate and access portions of the shared memory space for creating and manipulating the directory information. In connection with these embodiments, a system and related method for accessing directory information includes a computer network, a globally addressable data storage system, and a plurality of computers coupled to the network and the data storage system. The globally addressable data storage system provides persistent storage of data and contains directory information. The plurality of computers access the data storage system to obtain directory information. Based on the access patterns by the computers and/or the available network resources, the data storage system replicates and migrates directory information among two or more of the computers.

In one aspect, the invention relates to a method comprising the steps of providing a plurality of nodes interconnected by a network and storing on one or more of the nodes a directory containing both the directory information and information about the layout of the directory (i.e., where to find the directory information).

- 5 In another aspect, the invention relates to a method comprising the steps of providing a plurality of nodes interconnected by a network, providing a directory service on the network by installing on each of the nodes a directory service program that allows directory information to be accessed by each of the nodes and that stores on one or more of the nodes a directory including both the directory information and information about the location of the directory information on  
10 the network, and obtaining both the physical location of directory information of interest and the directory information of interest itself by accessing the directory service.

The foregoing and other objects, aspects, features, and advantages of the invention will become more apparent from the following description and from the claims.



Brief Description of the Drawings

In the drawings, like reference characters generally refer to the same parts throughout the different views. Also, the drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention.

5           FIG. 1 is a conceptual block diagram of a dynamic directory service according to the invention.

FIG. 2 is a diagram of the types of directory information that can be maintained by a dynamic directory service of the invention.

10           FIG. 3 is a diagram of a directory record used by a dynamic directory service according to the invention.

FIG. 4A is a flowchart of the steps a directory service of the invention takes to service a request from a network node for directory information.

FIG. 4B is a flowchart disclosing additional detail of the steps taken by the directory service to service the network node's request for directory information.

15           FIG. 5 is a diagram showing an example of directory records of a distributed dynamic directory service according to the invention.

Description

According to the invention, a computer network system 10 includes a plurality of network nodes that access a common directory service. The directory service is a specialized hierarchical structured database. Examples of the types of information that can be stored and maintained by the directory service include operating system configuration information, application program configuration information, network configuration information, network-accessible resources, exported devices and services, network printers, and network user account records. Each of the nodes on the network includes at least a directory service program that accesses and manages the directory service. The directory service may be stored in an addressable shared memory or it may be stored in a more traditional fashion. For example, each node may be responsible for storing a particular element or elements of the directory service. In such an embodiment, the directory service program can access a desired portion of the structured store using a globally-unique tag or identifier. The underlying system would translate the tag or identifier into one or more commands for accessing the desired data, including network transfer commands. In another embodiment, the directory service is stored in an addressable shared memory space, which allows the network nodes transparently to access portions of the structured store using standard memory access commands. In a preferred embodiment, the tag or identifier is an address into an addressable memory space such as a 128-bit address space.

The system 10 can be a file system, a database system, a Web server, an object repository system, or any other structured storage system that maintains an organized set of data. In the disclosed embodiment, the system 10 is a directory service that maintains various directory information.

Referring to FIG. 1, in one embodiment, the network system 10 includes a plurality of network nodes 12a-12d and a common directory service subsystem 20 that provides a dynamic directory service 22 according to the invention. The dynamic directory service 22 maintains and provides access to data including both the directory information itself and the physical layout of the directory on the network. Each of the nodes 12a-12d can include several sub-elements. For example, node 12a includes a processor 30a and a directory service program 32a for accessing the directory service 22. One or more of the nodes can include a monitor for displaying graphically (40, 42) the directory service 22.

A system 10 according to the invention can provide, among other things, each network node 12a-12d with shared control over the directory service 22 and, therefore, the system 10 can distribute control of the directory information across the nodes of the network. To this end, each node of the system 10, such as node 12a, includes a directory service program 32a that operates  
5 as a structured directory service adapted to maintain directory information and to employ all of the network nodes for storing and allowing access to the directory information. These cooperating elements provide a structured storage system that has a distributed architecture and thereby achieves greater fault tolerance, reliability, and flexibility than known directory services that rely on centralized control and one or more centralized servers. Accordingly, the invention  
10 provides computer networks with distributively controlled and readily scaled directory services.

Still referring to FIG. 1, in one embodiment, the system 10 maintains a directory service 22 within a globally addressable unstructured storage system. Each of the nodes 12a-12d can access that storage system and the directory service 22 through the directory service programs 32a-32d. At least a portion of the globally addressable unstructured storage system is supported  
15 by a physical memory system that provides persistent storage of data. For example, a portion of the storage system can be assigned or mapped to one or more hard disk drives that are on the network or associated with one or more of the network nodes 12a-12d as local hard disk storage for those particular nodes. Accordingly, FIG. 1 illustrates one possible system that provides the network nodes with access to a globally addressable unstructured storage system, wherein at  
20 least a portion of the storage space of that system is assigned to at least a portion of one or more of the persistent storage devices (e.g., hard disks) to allow the nodes addressably to store and retrieve data to and from the one or more persistent storage devices. The globally addressable storage system is described in the above-identified, incorporated-by-reference applications.

Each of the directory service programs 32a-32d is a software module that couples to the  
25 directory service. The directory service program 32a can stream data to, and collect data from, the directory service subsystem. Each of the directory service programs 32a-32d can be a peer incarnation (i.e., an instance) residing on a different one of the network nodes 12a-12d.

One or more of the directory service programs 32a-32d can provide a graphical user interface 42 that graphically depicts the directory service 22. The graphical user interface 42  
30 could allow a user at a node, for example at node 12a, to insert directory information graphically

within the directory service 22. To this end, the directory service program 32a can generate a set of commands that will present a stream of data that will result in directory information being stored within the directory service 22. As shown in FIG. 1, for node 12c only for simplicity, that node (which includes a graphical user interface 40) reflects the change to the directory service 22 affected by the directory service program 32a of the node 12a. In particular, the graphical user interface 40 of the node 12c can depict to a user that certain directory information is being placed within the directory service 22. As illustrated, a system user at node 12a can direct item 50a to be inserted at a set location within the directory service 22. The directory service program 32a then places the item 50a within the directory service 22 at the proper location. Moreover, node 12c detects the change within the directory service 22 and reflects that change within its graphical user interface 40.

A structured dynamic directory service according to the invention looks to all network nodes like a coherent, single directory service system when in fact it spans all participating nodes coupled to the network. The directory service of the invention differs from known directory services in a variety of ways. For example, the directory service of the invention: maintains data coherence among network nodes; automatically replicates directory information for redundancy and fault tolerance; automatically and dynamically migrates directory information to account for varying network usage and traffic patterns; and provides a variety of other advantages and advances, some of which are disclosed in the above-identified, incorporated-by-reference applications. The replication and migration can be done on the basis of node accesses and/or on the basis of the availability of network resources.

Referring to FIG. 2, a directory service according to the invention includes a structured store of data organized as a directory information set 66. The set 66 is a tree structure starting at a root 80 and ending at a leaf (for example, leaf 82). Each leaf represents a particular piece of directory information (e.g., user name or password). The set 66 thus is a collection of directory information organized hierarchically, for example as a tree structure or as a graph, rooted in the root 80. The non-leaf nodes in the tree (not including the root 80) are the entries 90, 92, 94, 100, 102, 200, 202, and 204, and the leaves in the tree are particular pieces of directory information 91, 82, 84, 86, 201, and 205-212 (e.g., a password or a network printer identifier) or empty entries. Sub-trees within a set can overlap by linking a leaf or a non-leaf to multiple entries.

The directory service of the invention can, as an option, employ more than one set. A benefit of breaking up the directory service 60 into a plurality of sets is that it may provide more flexible management for users of the directory service. As the directory service grows into very large sizes (e.g., hundreds of nodes with thousands of gigabits of storage), it may be desirable to have the directory information organized into groups of management entities such that management actions can be independently applied to individual groups without affecting the operation of the others.

In a set, the root (e.g., the root 80) provides the starting point to locate the directory information maintained by the directory service of the invention. The root can be, and preferably is, stored in a static and well-known location on the network (e.g., at a particular address or on one or more network nodes). When a node is accessing a set for the first time, it first looks up the root to determine the key associated with the set. Once it has determined the key, the node can access the root of the set. From the root, it then can traverse the set's entire tree to locate the desired piece of directory information.

As an example, in FIG. 2, a network node requesting the password of a user named Jones might pass the following string to the directory service: /USERS/JONES/PASSWORD. The directory service would then reference the set 66 and might find that the root 80 contains a list of all users, the non-leaf node 90 contains information about the user Jones, and the leaf 82 contains user Jones' password. In accordance with the invention, and as described more fully below, all of this information can reside physically on different network nodes and in fact can migrate from node to node and can be replicated on a plurality of network nodes.

FIGS. 3, 4A, 4B, and 5 further and more particularly illustrate the structure of a hierarchical distributed dynamic directory service according to the invention.

Referring to FIG. 3, a directory record 320 includes a record header 322 and one or more directory entries (two are shown, 324 and 326). Each directory entry includes a key field 330 and one or more data fields 332. The key field 330 could be, for example, "company name," and the associated data field 332 could be "Acme Corporation." Note that for GUID tree records (described below), there preferably are two data fields, namely "GUID of next record" identifying the identifier of the next record to be referenced by the directory service and "GUID responsible node" identifying the node responsible for the "GUID of next record." The key field 330 could

also be a globally unique identifier (GUID). The data field 332 can include actual directory information (for example, "Acme Corporation"), information about where to go to locate the directory information of interest or additional information about where to go to locate it (for example, a globally unique identifier or GUID), or information about what nodes have copies of the directory information of interest.

In one particular embodiment, directory record 320 can be a page of a global address space that spans both persistent (e.g., hard disks) and volatile (e.g., RAM) storage devices. For example, the page can be a 4 kilobyte portion of the shared address space described in the above-identified, incorporated-by-reference applications. In this embodiment, the GUIDs are unique addresses of the global address space such as 128-bit addresses in a huge  $2^{128}$  address space.

Referring still to FIG. 3, each directory record 320 includes a record header 322 that includes attribute information for that record, and that attribute information typically is metadata for the directory record. The record 320 further includes one or more directory entries, such as the depicted directory entries 324 and 326, that provide an index into a portion of the directory service (non-leaf entries) or actual directory information (leaf entries). Accordingly, the non-leaf directory entries subdivide the directory information maintained by the directory service of the invention. For example, if the directory entries 324 and 326 are entries of a GUID tree record, they can subdivide the directory into two sub-portions, with the first portion referencing one-half of the directory and the second portion referencing the other half of the directory. Accordingly, in this example, the directory entry 324 provides an index for half of the directory, and in complement thereto the directory entry 326 provides an index for the other half. Continuing with this example, each of these directory entries 324, 326 can, via the data fields, point to other records and responsible nodes in which directory entries have data fields that point to still other records and responsible nodes and so on until the actual directory information of interest is located in some data field of some entry of some record by the directory service (a so-called leaf entry). The directory service then provides that directory information to the requesting node. In this way, the directory service of the invention provides both (1) location information for directory information maintained by the directory service and (2) the actual directory information itself, all in the same directory structure.

In accordance with the invention, directory records are indexed by globally unique identifiers (GUIDs), such as 128-bit values. In the directory records, these GUIDs can appear in the data fields or the key fields of various of the directory entries. As described in more detail below, a GUID subtree within the directory provides a map or assignment of GUIDs to network nodes that have a copy of certain directory information. It is the leaf nodes of the GUID subtree that identify the one or more nodes that have a copy of desired directory information. Once these nodes are identified by the directory service, the directory service accesses one or more of them (typically just one of the identified nodes) to obtain the directory information and then pass it to the requesting node. Optionally, the directory system can choose which node to contact to obtain the directory information based on the state of the global system such as the current load on the various nodes or the quality of the network service between the local node and the node(s) that have copies of the desired directory record.

Referring to FIG. 4A, to obtain certain desired directory information (e.g., the password of a user named Jones), a network node passes an appropriate key to the directory service (step 500). For example, the node might pass the following key to the directory service: /USERS/JONES/PASSWORD. The directory service receives the key and either locates the desired directory information and passes it to the requesting node or returns an error if the desired directory information cannot be found (step 600). More specifically, in attempting to satisfy the request from the node, the directory service of the invention first determines if the key corresponds to directory information that is stored on the node itself in, for example, the node's RAM or on the node's hard disk (step 602). In the disclosed embodiment, the directory service performs step 602 by accessing a lookup table (such as a hash table) maintained by each of the network nodes. This table identifies the directory records that are locally cached on the node. If the requested directory information is stored locally on the requesting node, the directory service retrieves the requested directory information from the node's local cache and then provides that directory information to the node (step 604). If the requested directory information is not stored locally on the requesting node, the directory service recursively invokes itself to access the directory record(s) that have the GUID tree structure, and the directory service descends or walks the GUID tree by accessing the various records until the desired directory entry is located. The leaf node in the GUID tree contains the mapping from GUID to the set of nodes caching the record associated with that GUID. The directory service uses this information to select a node

from which it acquires a copy of the desired record (step 606). The directory service then obtains the requested directory information and provides it to the node (step 608).

It is important to note that, while traversing the GUID tree, the directory service first always checks to see if entries indexed by a GUID are cached locally before the directory service  
5 invokes recursively the GUID tree to locate a copy of the record remotely.

At this point, the directory service can store a copy of (replicate) this directory information (e.g., user Jones' password) on the requesting node. Actually, if replication is performed by the directory service, all or a portion of the record that contains this directory information is replicated on the requesting node. If the directory service decides to replicate this  
10 record on the requesting node, the directory service adds the requesting node's identifier (e.g., a number) to the leaf node in the GUID tree that stores the list of nodes caching the record associated with the GUID, and the directory service then updates or invalidates the local caches of all of the other network nodes that also have that directory record cached locally.

In one embodiment, the directory service makes the decision on whether or not to  
15 replicate based on node access patterns and/or on the availability of certain network resources. For example, if a node has requested a certain record a certain number of times, the directory service will replicate that record on that node. The directory service monitors and records a variety of network information, including node access patterns and network resource availability, and some or all of this network information is used to determine what records should be  
20 replicated on what nodes. In accordance with the invention, this replication feature of the directory service results in certain records being replicated on the nodes that most often access those records and not being replicated on (and/or removed from) nodes that access those records less often or not at all. The directory service of the invention provides this replication feature dynamically during normal operation, and thus, as access patterns change, the records and the  
25 copies of the records move or migrate among the various network nodes to accommodate the changing patterns. This replication/migration feature of the invention points up an important aspect of the invention, and that is that all of the directory records are homeless (except possibly the root directory record which, while it typically will be replicated, typically is placed at fixed locations on the network).



Because the directory records contain, according to the invention, both the GUID tree structure (i.e., the information about where directory information is located) and the actual directory information, the directory structure (i.e., the GUID tree) is dynamically moved and replicated among the network nodes just as is the directory information itself.

5        These dynamic replication and migration features provide tremendous benefits to networks utilizing the directory service of the invention. For example, networks utilizing the directory service of the invention can be expanded (i.e., additional nodes can be added to the network and/or other networks can be interconnected to the network) without substantially impacting the performance and speed realized by the nodes as they access directory information  
10        via the directory service. That is, the directory service scales very well. This is because, once the network is up and running, all nodes generally will have cached locally the records they most often access, and access times for these nodes for the directory information that is cached locally will be the same (very fast) regardless of the size of or the traffic on the network.

Referring to FIG. 4B, the recursive invocations identified in step 606 (FIG. 4A) are now  
15        described in more detail. Step 606 includes the following substeps that are performed by the directory service. The directory service first locates the root directory record (step 610), and it indexes into that record using the key (e.g., /USERS/JONES/PASSWORD) passed to it by the requesting node. The directory service then checks to see if a portion of the key (e.g., /USERS) matches a directory entry in the root directory (step 612). If it does not match, the directory  
20        service returns an error message to the requesting node (step 614). If there is a corresponding entry in the root directory, the directory service next determines if that directory entry contains the requested directory information in its data field (step 616). If so, the directory service retrieves the requested directory information from that data field and provides it to the requesting node (618).

25        Referring now to FIGS. 4B and 5, a root directory record 400 could be the root record referred to in step 612. If directory 402 contains the requested directory information in its data field (step 616), the directory service need only to index into that entry 402 using the key and retrieve the requested directory information from the data field. Note that, as indicated in FIG. 5, the root directory record 400 could be replicated (401, 403, 405) on three other network nodes  
30        in accordance with the replication feature of the directory service of the invention. In fact, it may

be that the root directory record 400 is replicated on all network nodes. This could be reasonable in light of the fact that every network node will likely refer to the root directory record when the node first invokes the directory service of the invention, and frequently thereafter.

5        If the entry in the root directory record does not contain the requested directory information in the entry's data field (step 616 of FIG. 4B), the directory service determines whether the entry's data field contains a GUID (step 620). If it does contain a GUID, this means that the GUID tree must be navigated by the directory service to located the requested directory information. The directory service thus follows the GUID in the root directory record's entry's  
10   data field to a new directory record (step 628), possibly the GUID tree root directory record (404 in FIG. 5). The GUID thus is a logical link to a record that will provide more information about the GUID tree structure and ultimately allow the directory service to locate the requested directory information.

15        As shown in FIG. 5, the GUID in the root directory record's entry's data field can be a logical link or pointer to a record that is the GUID tree root directory record 404. As with the root directory record 400, the GUID tree root directory record 404 is likely to be replicated on many or all of the network nodes. In the example of FIG. 5, two replicas (405, 407) of the record 404 are depicted.

20        The loop defined by steps 628, 630, and 634 in FIG. 4B illustrates the recursive nature of the directory service. That is, once the GUID tree records are entered via the GUID tree root directory record 404 (FIG. 5), and if the data fields of the appropriate directory entries in the GUID tree records contain GUIDs, the directory service will enter a recursive loop whereby it accesses a plurality of records (identified by the ellipsis 410 in FIG. 5) until it reaches a record that has a directory entry with one or more nodes listed its data field (step 622). Such a record is  
25   a leaf of the GUID tree structure, and is indicated as record 412 in FIG. 5. As with all records maintained by the directory service of the invention, this record 412 can be replicated (413) on at least one other network node. Also, while not presently preferred, in general it is possible at any point during this recursive GUID tree walk to encounter a directory entry having the requested directory information in its data field (step 630 of FIG. 4B), and in such a case the directory

service would retrieve from the data field the requested directory information and provide it to the requesting node (632).

It is important to note that the cache of locally stored directory entries can be indexed so that at any stage of the GUID tree walk operation it is possible to obtain the desired record from the local cache instead of the tree structure. For example, if record 398 is cached on the node  
5 attempting to lookup "/USERS/JONES/PASSWORD," the directory service will be able to find the locally cached record via a hash lookup for either /USERS/JONES/PASSWORD or the corresponding GUID. Similarly, while traversing the GUID subtree, the directory service first always checks to see if entries indexed by a GUID are cached locally before the directory service  
10 invokes recursively the GUID tree to locate a copy of the record remotely.

In the course of the recursion, the various records that the directory service accesses lead the directory service to the leaf record 412. This happens by the directory service beginning at the GUID tree root directory record 404 and examining the key field of each of the directory entries in the record 404 (or one of the copies 405, 407 of the record 405 on other nodes in the  
15 network) to determine the range in which the GUID logical link from the record 400 falls. If the GUID logical link from the record 400 falls within the range identified by the key field of directory entry 409, the GUID in the data field of the entry 409 is used as the logical link or pointer to the next record. This process continues until the ranges in the key fields of the directory entries of the subsequent records reduce to a single GUID such as the GUID "12540"  
20 in the key field of a directory entry 411 in the GUID leaf record 412.

Again, while traversing the GUID subtree, the directory service first always checks to see if entries indexed by a GUID are cached locally before the directory service invokes recursively the GUID tree to locate a copy of the record remotely. In referencing the GUID tree, the directory service uses the responsible node information in the data fields of the GUID tree  
25 records in the location process.

It is important to realize that all of these records that the directory service accesses in this recursive loop (and, in general, even when it is not in the recursive loop) can, and in many instances will, be located on different network nodes. This is because the directory service of the invention is distributed, and in general the records that make up the directory and that contain the  
30 various pieces of the directory information maintained by the directory service are located on

different nodes all over the network. As the GUID tree is walked in the manner described herein, the directory service can replicate the accessed GUID tree records and make them local on the requesting node. This replication is optional and is accomplished as described hereinabove.

It also is important to realize that the records that describe the structure of the directory layout (i.e., the GUID tree records) are stored by the directory service in the same manner as it stores the records having the actual directory information and also maintained by the directory service in the same way that it maintains the records with the actual directory information.

Referring to step 622 of FIG. 4B and record 412 of FIG. 5, the data field in the entry 411 of the GUID leaf record 412 contains a list of nodes. This is a list of nodes that have a copy of the record that contains the requested directory information. The directory service can now access any one of those listed nodes to obtain the requested directory information (step 624).

Instead of going from the root directory record 400 (or any one of its copies 401, 403, 405) to walking the GUID tree to obtain the requested directory information, it is possible that the directory service of the invention will find an entry in the root directory record 400 that provides a logical link (e.g., GUID "12540") in its data field directly to an entry 399 in a record 398 that has the requested directory information in its data field. It also is possible, as mentioned hereinabove, that an entry 402 in the root directory record 400 itself will have the requested directory information in its data field. A typical situation, however, is when the GUID tree is walked by the directory service via the GUID tree records in order to locate and return the requested directory information to the requesting network node.

In accordance with the invention, the directory service tracks ownership and responsibility for directory information thereby providing a level of indirection between the actual directory information itself and the physical location of that directory information on the network.

It should be appreciated that the directory service of the invention comprises a hierarchical structured storage mechanism for directory information. To this end, the directory service of the invention provides a structure that continually subdivides itself into smaller and smaller sections. Further, each section is represented by directory records of the same structure, but each section indexes different amounts (sizes) of directory information.

In accordance with the invention, more frequently accessed directory information is copied and distributed among various network nodes, and rarely used directory information generally will appear on only a few network nodes. Also, directory records will migrate to those nodes that access them most, providing a degree of self-organization that reduces network traffic.

5        In general, all of the functionality of the directory service of the invention can be implemented in software. In one embodiment, an instance of a directory service program resides and executes on each of the network nodes and provides all of the directory service features and functionality described herein. It is possible to perform one or more of the various functions of the directory service with dedicated electronics or a combination of hardware and software, and  
10    this hardware might be added to a general purpose computer to implement the directory service of the invention.

Variations, modifications, and other implementations of what is described herein will occur to those of ordinary skill in the art without departing from the spirit and the scope of the invention as claimed. Accordingly, the invention is to be defined not by the preceding illustrative  
15    description but instead by the spirit and scope of the following claims.

What is claimed is:

Claims

1. A method for obtaining directory information, comprising:  
providing a plurality of nodes interconnected by a network;  
5 providing a directory service on the network by installing on each of the nodes a directory service program that allows directory information to be accessed by each of the nodes and that stores on one or more of the nodes a directory that includes both the directory information and information about the location of the directory information on the network; and  
obtaining both the location information and the directory information by accessing the  
10 directory service.
2. The method of claim 1 wherein the directory service providing step further comprises installing the directory service program on each of the nodes and the directory service program automatically replicates directory information.
3. The method of claim 2 wherein the directory service providing step further comprises  
15 installing the directory service program on each of the nodes and the directory service program automatically replicates directory information based on accesses of the directory service by the nodes.
4. The method of claim 2 wherein the directory service providing step further comprises installing the directory service program on each of the nodes and the directory service program  
20 automatically replicates directory information based on available resources on the network.
5. The method of claim 1 wherein the directory service providing step further comprises installing the directory service program on each of the nodes and the directory service program automatically migrates directory information.
6. The method of claim 5 wherein the directory service providing step further comprises  
25 installing the directory service program on each of the nodes and the directory service program automatically migrates directory information based on accesses of the directory service by the nodes.

7. The method of claim 5 wherein the directory service providing step further comprises installing the directory service program on each of the nodes and the directory service program automatically migrates directory information based on available resources on the network.

8. A method for storing directory information, comprising:

5 providing a plurality of nodes interconnected by a network; and  
storing on one or more of the nodes a data structure representative of both the directory information and information about the location of the directory information on the network.

9. A system for accessing directory information, comprising:

10 a computer network;  
a storage system for maintaining directory information; and  
a plurality of computers, coupled to the network and the storage system, for accessing the storage system to obtain directory information;  
wherein the storage system replicates and migrates directory information among two or more of the computers.

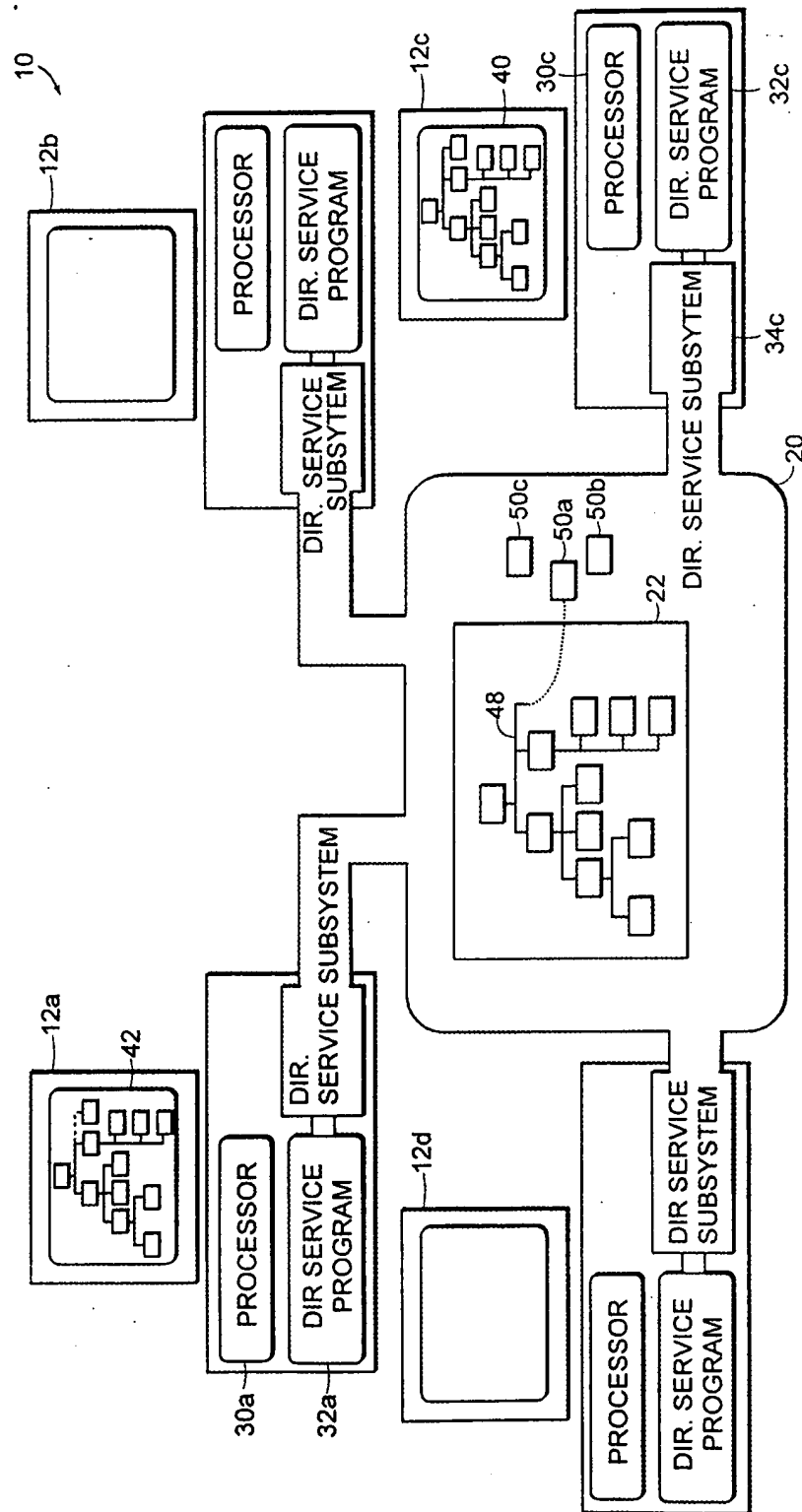


FIG. 1



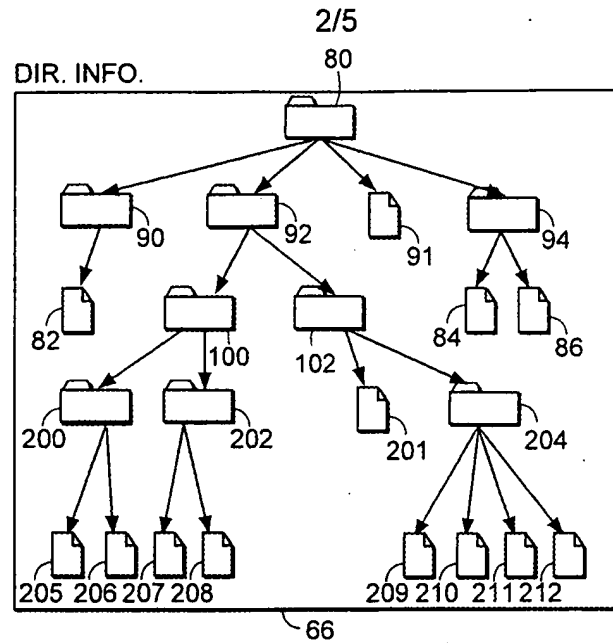


FIG. 2

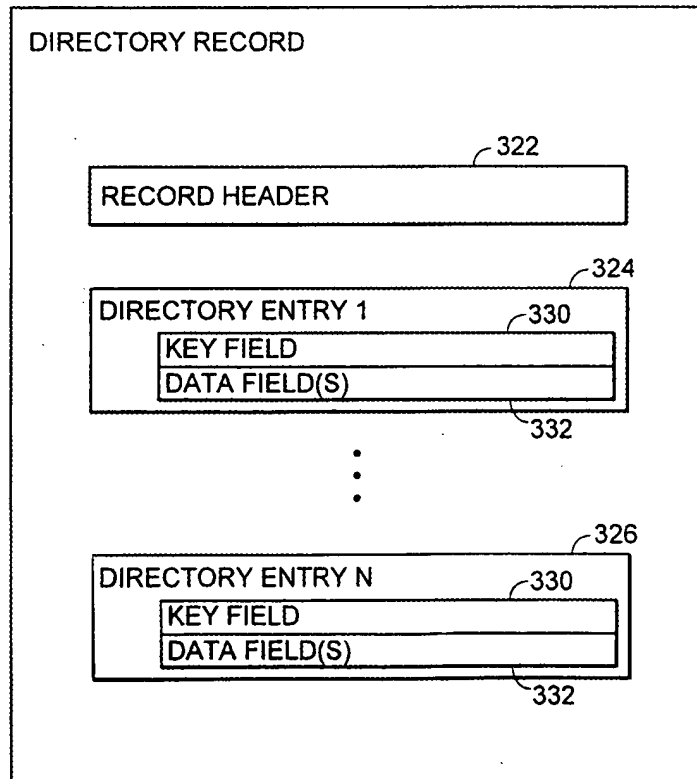


FIG. 3

3/5

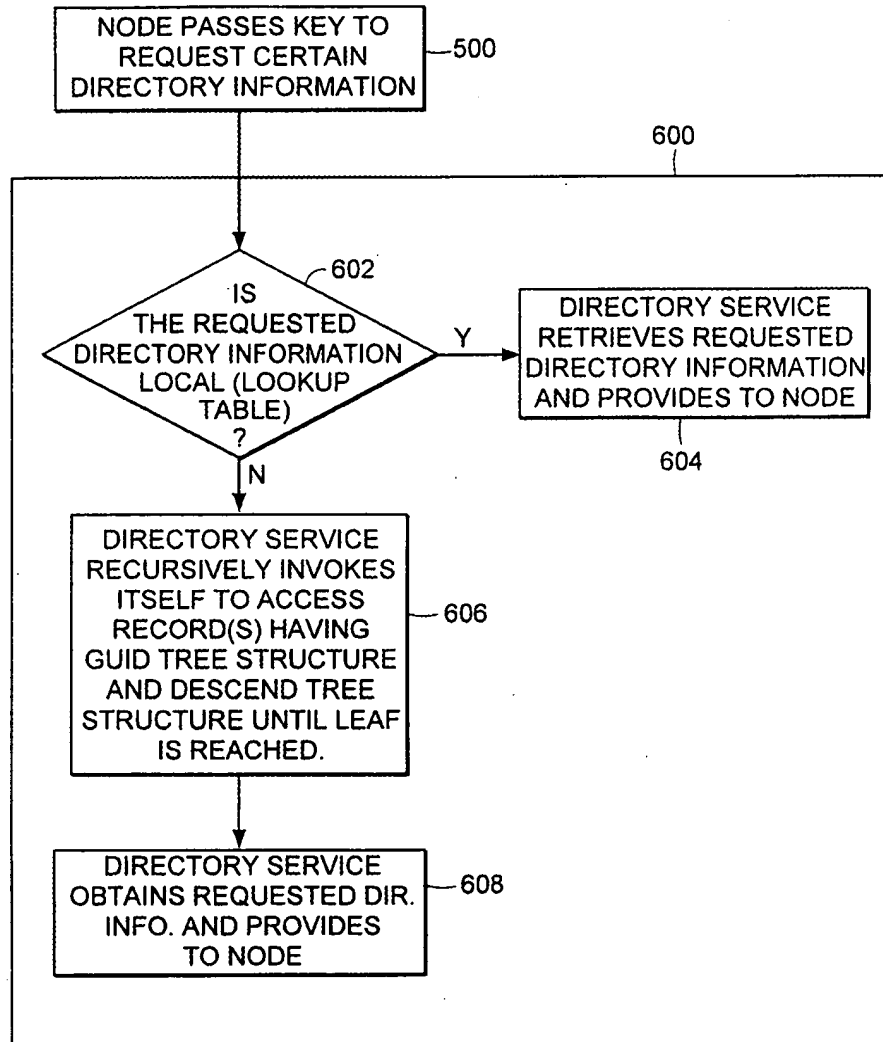


FIG. 4A

SUBSTITUTE SHEET (RULE 26)

4/5

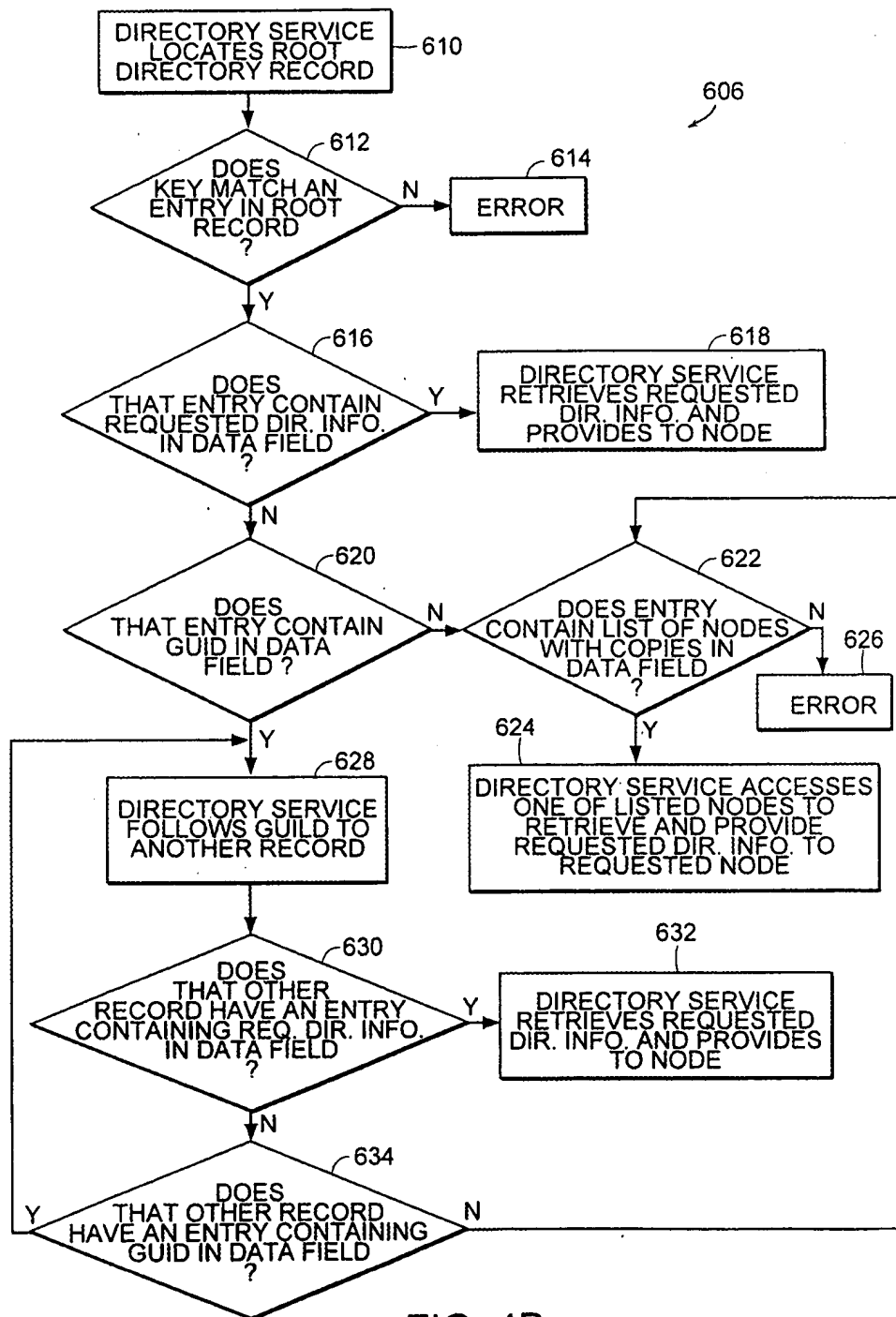


FIG. 4B

SUBSTITUTE SHEET (RULE 26)

5/5

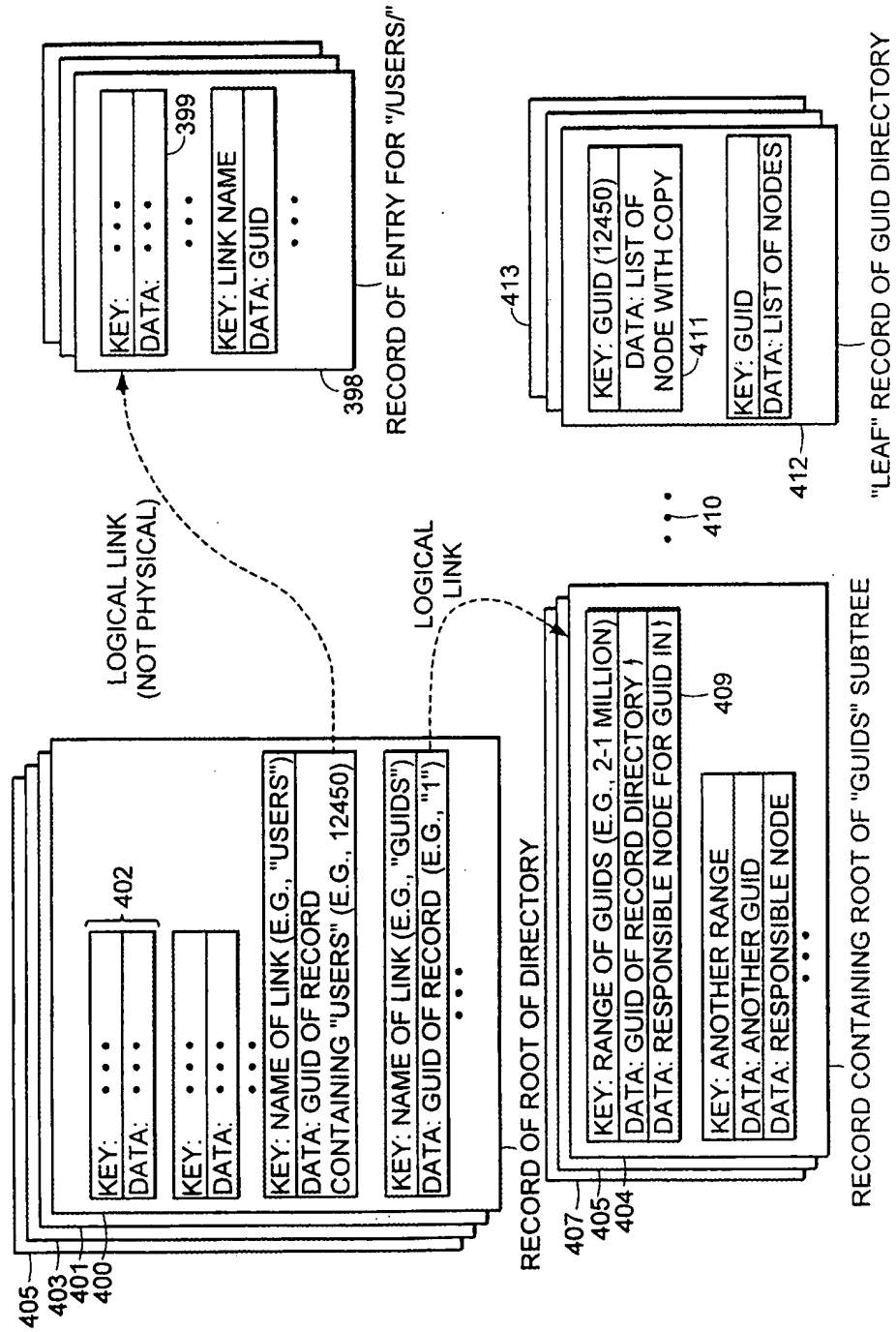


FIG. 5

# INTERNATIONAL SEARCH REPORT

Int. Application No  
PCT/US 97/21733

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 6 G06F17/30

According to International Patent Classification(IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 95 22111 A (ERICSSON TELEFON AB L M) 17 August 1995	1,9
A	see abstract see claims 1,2; figure 3	2-8,10
X	EP 0 661 651 A (MICROSOFT CORP) 5 July 1995	10
A	see column 5, line 52 - column 10, line 7; figures 1,6	1-9
A	WO 96 30847 A (TELE COMMUNICATIONS INC) 3 October 1996 see the whole document	1-10
A	WO 95 14279 A (CONNER PERIPHERALS INC) 26 May 1995 see the whole document	1-10

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

### \* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

9 April 1998

Date of mailing of the international search report

23/04/1998

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Michel, T

# INTERNATIONAL SEARCH REPORT

Information on patent family members

Int. . . . . tional Application No

PCT/US 97/21733

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9522111 A	17-08-95	AU 687168 B	19-02-98
		AU 1723695 A	29-08-95
		CN 1140500 A	15-01-97
		EP 0744055 A	27-11-96
		FI 963107 A	07-08-96
		JP 9508734 T	02-09-97
		NO 963281 A	08-10-96
		SE 9400410 A	09-08-95
EP 0661651 A	05-07-95	US 5689700 A	18-11-97
		CA 2138623 A	30-06-95
		JP 7210442 A	11-08-95
		US 5649194 A	15-07-97
		US 5675787 A	07-10-97
WO 9630847 A	03-10-96	AU 5524096 A	16-10-96
		EP 0819278 A	21-01-98
WO 9514279 A	26-05-95	US 5495607 A	27-02-96
		AU 1093695 A	06-06-95
		EP 0729618 A	04-09-96
		JP 9509268 T	16-09-97
		US 5678042 A	14-10-97